

WHAT IS CLAIMED IS:

- 1 1. A method for balancing a workload for a plurality of processors in
2 a multiple processor computer system, the system designed for processing a
3 plurality of packets from a plurality of connections, the method comprising:
4 assigning a packet to a hash bucket determined by performing a
5 predetermined hash function; and
6 queuing the hash bucket to a processor so that the workload of all the
7 processors are balanced,
8 wherein the hash function relates to a predetermined number of hash
9 buckets, and wherein a plurality of packets from different connections can be
10 assigned to the same hash bucket.
- 1 2. The method of claim 1 wherein the step of assigning includes a step
2 of using a source address, source port, destination address, and destination port
3 parameters from the header section of the packet to identify the hash bucket.
- 1 3. The method of claim 1 wherein the hash function is designed to
2 assign a plurality of packets from a connection to the same hash bucket.
- 1 4. The method of claim 1 further comprising monitoring the workload
2 of each processor involved in the system.

1 5. The method of claim 1 wherein the step of queuing further
2 includes:
3 identifying statistically a percentage of capacity usage for each processor;
4 obtaining imbalance distribution index for each processor from the
5 identified percentage of capacity usage;
6 distributing the hash buckets in proportion to the imbalance distribution
7 index and in reverse proportion to a total imbalance index,
8 wherein the total imbalance index is a summation of all obtained
9 imbalance distribution indexes.

1 6. The method of claim 1 wherein the step of queuing further includes
2 a step of applying a queuing model for packets in the hash bucket to assure that
3 packets from a particular connection do not unduly occupy the processor for an
4 undesirable time period.

1 7. A computer program for balancing workload for a plurality of
2 processors in a multiple processor computer system, the system designed for
3 processing a plurality of packets from a plurality of connections, the computer
4 program comprising:
5 means for assigning a packet to a hash bucket determined by performing a
6 predetermined hash function; and
7 means for queuing the hash bucket to a processor so that the workload of
8 all the processors are balanced,
9 wherein the hash function has a predetermined number of hash buckets,
10 and wherein a plurality of packets from different connections can be assigned to
11 the same hash bucket.

1 8. The computer program of claim 7 wherein the means for assigning
2 includes means for using a source address, source port, destination address, and
3 destination port parameters from the header section of the packet to identify the
4 hash bucket.

1 9. The computer program of claim 7 wherein the hash function is
2 designed to assign a plurality of packets from a connection to the same hash
3 bucket.

1 10. The computer program of claim 7 further comprising means for
2 monitoring the workload of each processor involved in the system.

1 11. The computer program of claim 7 wherein the means for queuing
2 further includes means for:
3 identifying statistically a percentage of capacity usage for each processor;
4 obtaining imbalance distribution index for each processor from the
5 identified percentage of capacity usage;
6 distributing the hash buckets in proportion to the imbalance distribution
7 index and in reverse proportion to a total imbalance index,
8 wherein the total imbalance index is a summation of all obtained
9 imbalance distribution indexes.

1 12. The computer program of claim 7 wherein the means for queuing
2 further includes means for applying a queuing model for packets in the hash
3 bucket to assure that packets from a particular connection do not unduly occupy
4 the processor for an undesirable time period.

1 13. A method for processing a plurality of connections with a plurality of
2 timer threads by a plurality of computer processors in a multiple processor
3 computer system, the method comprising:
4 providing a plurality of hash buckets related to a hash function;
5 mapping a connection to one of the hash buckets; and
6 assigning each hash bucket to a processor timer thread based on a
7 workload thereof so that the processor only processes the connection mapped to
8 the assigned hash bucket,
9 wherein a plurality of timer threads for the computer processors thus
10 process a plurality of connections simultaneously.

1 14. The method of claim 13 further comprising monitoring the
2 workload of the each processor.

1 15. The method of claim 13 wherein the step of mapping is based on four
2 connection parameters, which are local address, local port, remote address, and
3 remote port.

1 16. A computer program for processing a plurality of connections with a
2 plurality of timer threads by a plurality of computer processors in a multiple
3 processor computer system, the program comprising:
4 a plurality of hash buckets related to a predetermined hash function;
5 means for mapping a connection to one of the hash buckets; and
6 means for assigning each hash bucket to a processor timer thread based on
7 a workload thereof so that the processor only processes the connection mapped
8 to the assigned hash bucket,
9 wherein a plurality of timer threads for the computer processors thus
10 process a plurality of connections simultaneously.

1 17. The program of claim 16 further comprising means for monitoring
2 the workload of the each processor.

1 18. The program of claim 16 wherein the means for mapping is based
2 on four connection parameters, which are local address, local port, remote
3 address, and remote port.